NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

TRACK SPACING FOR AN ARCHIMEDES SPIRAL SEARCH BY A MARITIME PATROL AIRCRAFT (MPA) IN ANTI-SUBMARINE WARFARE (ASW) OPERATIONS

by

Byungsoo Son

December 2007

Thesis Advisor:	James Eagle
Second Reader:	Paul Sanchez

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** December 2007	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE: Track Spacing for An Archimedes Spiral Search by A Maritime Patrol Aircraft (MPA) in Anti-Submarine Warfare (ASW) Operations			**5. FUNDING NUMBERS**
6. AUTHOR(S) Byungsoo Son			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			**12b. DISTRIBUTION CODE**
13. ABSTRACT (maximum 200 words)			
14. SUBJECT TERMS Anti-Submarine Warfare(ASW), Archimedes spiral, Datum Search, Random Tour, Central Limit Theorem, Rayleigh, NOLH(Nearly Orthogonal Latin Hypercube), Simulation Experiment			**15. NUMBER OF PAGES** 49
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

The military threat of hostile submarines is increasing and the need for effective Anti-Submarine Warfare (ASW) operations is also increasing. In response, the ROK government and military have improved their ASW capabilities. In this thesis, the recommended track spacing for an Archimedes spiral search in a datum search problem was studied. To find a recommended track spacing, three analytical approaches were explored. Each of three analytical approaches has its own strengths and weaknesses. This analysis combined three analytical functions into a single parameterized expression. To find the best-fit parameters maximizing probability of detection, a simulation experiment with a NOLH (Nearly Orthogonal Latin Hypercube) design was used.

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

TRACK SPACING FOR AN ARCHIMEDES SPIRAL SEARCH BY A MARITIME PATROL AIRCRAFT (MPA) IN ANTI-SUBMARINE WARFARE (ASW) OPERATIONS

Byungsoo Son
Captain, Korean Army
B. E., Korea Military Academy, 2003

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
December 2007

Author: Byungsoo Son

Approved by: James Eagle
Thesis Advisor

Paul Sanchez
Second Reader

James Eagle
Chairman, Department of Operations Research

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The military threat of hostile submarines is increasing and the need for effective Anti-Submarine Warfare (ASW) operations is also increasing. In response, the ROK government and military have improved their ASW capabilities. In this thesis, the recommended track spacing for an Archimedes spiral search in a datum search problem was studied. To find a recommended track spacing, three analytical approaches were explored. Each of three analytical approaches has its own strengths and weaknesses. This analysis combined three analytical functions into a single parameterized expression. To find the best-fit parameters maximizing probability of detection, a simulation experiment with a NOLH (Nearly Orthogonal Latin Hypercube) design was used.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to express my gratitude to those who contributed to the successful and timely completion of this thesis.

First, I would like to thank my advisor, Professor James Eagle for his patience, guidance and constant support. I would like also to thank my second reader, Professor Paul Sanchez for his expertise, cooperation and willingness reading this thesis. Other members of the NPS faculty also have my appreciation for their excellent instruction.

My deepest gratitude also goes to the Korea National Defense University, especially Professor Jae-young Lee who encouraged and supported me for last one year.

Finally, I would like to thank my family (parents, sisters, and brother), fiancée, and friends for supporting me.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. BACKGROUND

The Republic of Korea (ROK) is surrounded by several powerful countries and also by water on three sides. For several decades, the ROK has faced an increasing threat of hostile submarines. In response, the government and military have improved their anti-submarine warfare (ASW) capabilities. Weapon systems such as the P-3C maritime patrol aircraft (MPA) have been deployed.

These ASW systems are important for preventing terrorist invasions by submarine and to protect the sea lanes. This thesis will develop an ASW search tactic for an MPA aircraft searching for a patrolling submarine.

B. SCENARIO

The periscope, communications antenna, or snorkel of a target submarine is detected by the long-range, surface search ASW radar on an MPA aircraft. The initial radar detection provides an estimated target position, called datum. The aircraft then flies directly to datum and conducts a spiral-out ASW search for the target. The target submarine submerges below periscope depth immediately after radar detection and continues its underwater patrol.

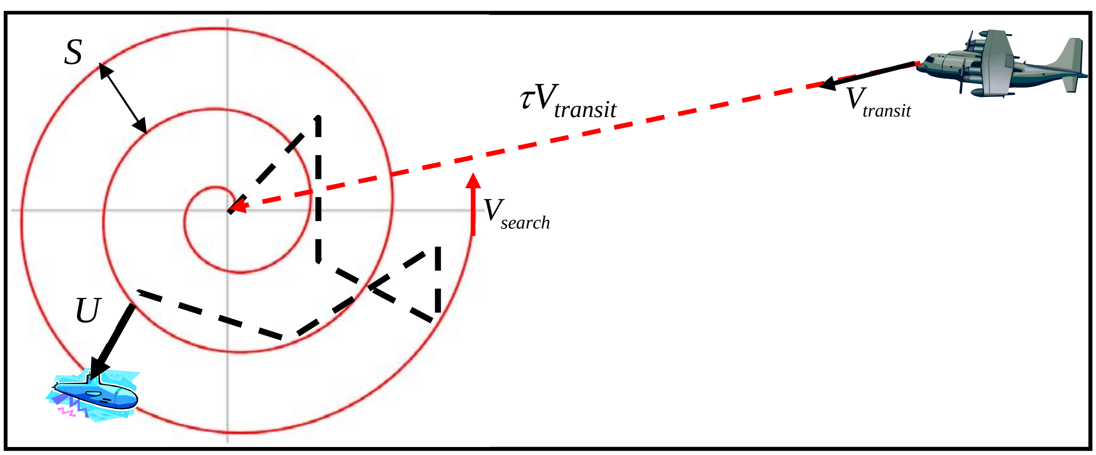


Figure 1. ASW Search Scenario

C. RESEARCH QUESTION

Referring to Figure 1, the problem parameters are:

U : target speed (kt)

V : aircraft speed (kt)

R : range at which the aircraft can detect the submarine (nm)

τ : time required for aircraft to fly from where detection occurred to datum (hr)

λ : the Poisson rate at which the target changes course (hr^{-1})

The main research question investigated here is as follows: Given problem parameters U, V, R, τ , and λ , what is the aircraft search track spacing S^* which maximizes the probability of detection of the submarine by the ASW aircrafts?

D. STRUCTURE OF THESIS

Chapters II and III discuss the mathematical model of a target submarine and a searcher aircraft which conducts an ASW search operation. Chapter IV develops three analytical approaches to suggest good track spacings for the spiral-out search. Chapter V develops the design of the simulation experiment and the analysis for constructing a meta-model to find the recommended track spacing. Chapter VI discusses the simulation results. Chapter VII provides conclusions and recommendations for further research.

II. SEARCHER MOTION MODEL

A. ARCHIMEDES SPIRAL SEARCH

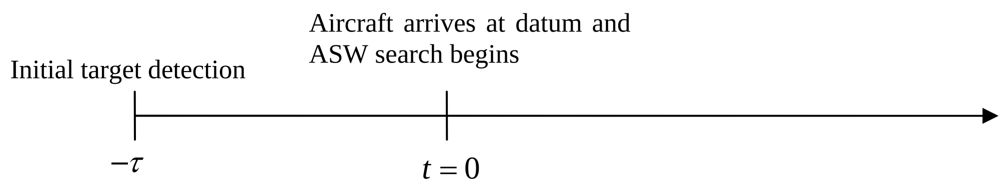


Figure 2. Search Timeline

The target is initially detected at time $t = -\tau$. At time 0, the searching aircraft arrives at datum and begins a spiral-out search with constant track spacing *S* and speed *V*. The aircraft carries a detector with certain detection range *R*.

The aircraft track will follow an Archimedes Spiral [1]. For such a search path, the radial distance from datum in terms of the angular position θ in radians is

$$r(\theta) = \frac{\theta S}{2\pi}.$$

So when θ increases by 2π (one revolution), the radial distance increases by *S*.

In Chapter IV, it is shown that at time $t \geq 0$, the approximate radial position at time t of a searcher with speed *V* is

$$r(t) = \sqrt{\frac{SVt}{\pi}}.$$

THIS PAGE INTENTIONALLY LEFT BLANK

III. TARGET MOTION MODEL

A. RANDOM TOUR

This model assumes the target's motion is a random tour [2]. The target's motion is assumed to be independent of the search, and not specifically directed toward escape. The target may move a considerable distance from the point of initial detection (datum) as the search for it proceeds. The target motion can be beneficial for the searcher in that it increases relative speed, but it can also be harmful as it increases the size of the area in which the target can be located.

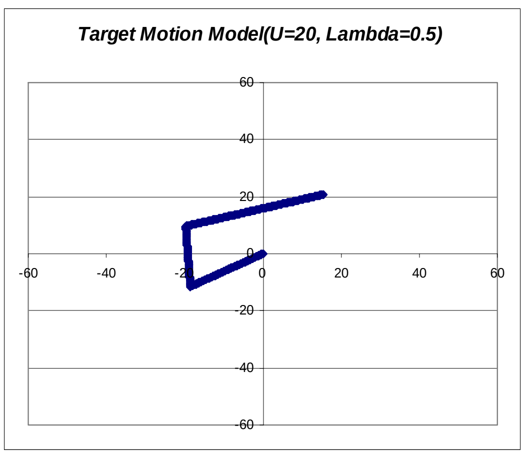


Figure 3. Description of Random Tour Target

B. PROBLEM PARAMETERS

Figure 3 shows an example of a random tour target motion with λ of .5 course changes/hr, and U of 20 nm/hr.

1. U (Target Speed)

Target speed U is assumed to be constant.

2. λ (Course Change Rate)

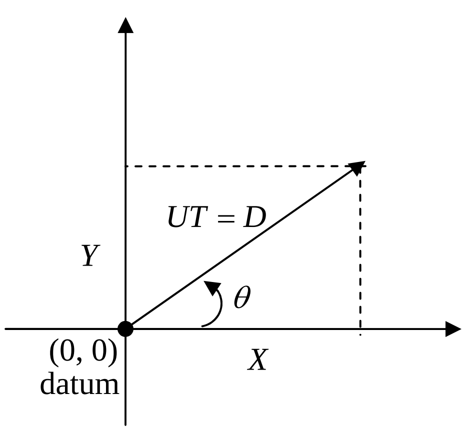
The target changes its direction at times determined by a Poisson Process [3] with rate λ. For instance, if λ=2 / hour, the target changes its course two times per hour on the average, and the time between course changes is exponentially distributed with mean 1/2 hour.

3. σ (Observation Error)

Immediately after the initial radar detection at time $t = -\tau$, the target's location distribution is circular bivariate normal with equal means (0, 0) and variances $\left(\sigma^2, \sigma^2\right)$. In this thesis, we will assume $\sigma = 1$ nm, consistent with a long-range radar detection of a small target.

C. NORMAL APPROXIMATION FOR THE PURE RANDOM TOUR TARGET

For large enough time t, the Central Limit Theorem [4] guarantees that the target position will be approximately circular bivariate normal.



U : target speed

T : duration of 1st leg (exponentially distributed with mean $1/\lambda$)

θ : Target's course during 1st leg (uniformly distributed between 0 - 2π radians)

D : length of 1st leg (exponentially distributed with mean U/λ)

Figure 4. The Initial Random Tour Leg

To see this, consider the following development. Since target's course change rate is λ, the time to the first course change, T, is an exponentially distributed random variable with rate λ. We write this as $T \sim \exp(\lambda)$. So,

$$P(D \le d) = P(UT \le d)$$

$$= P\left(T \le d/U\right)$$

$$= 1 - \exp\left(-\lambda d/U\right).$$

Thus, $D \sim \exp\left(\lambda/U\right)$. That is D, the distance travelled by the target during the first leg, is an exponentially distributed random variable with mean U/λ.

Now, we want to find the expected value and variance of the target position after one leg. Each new course θ is uniformly distributed. So,

$$\theta \sim U(0, 2\pi), \text{ and}$$

$$X = D\cos\theta.$$

Now by symmetry of the cosine function,

$$E(X) = 0.$$

So,

$$V(X) = E(X^2) - (E(X))^2$$

$$= E(X^2).$$

Now conditioning on the target course θ,

$$E(X^2) = \int_{\tau=0}^{2\pi} E(X^2 | \theta = \tau) f_\theta(\tau) d\tau$$

$$=\left(\frac{1}{2\pi}\right)\int_{\tau=0}^{2\pi} E\left(D^2\cos^2\tau\right)d\tau$$

And by the symmetry of $\cos^2(\tau)$,

$$=\left(\frac{1}{\pi}\right)\int_{\tau=0}^{\pi} \cos^2\tau E\left(D^2\right)d\tau .$$

And from the properties of the exponential distribution,

$$=\left(\frac{1}{\pi}\right)\int_{\tau=0}^{\pi} \left(\frac{2U^2}{\lambda^2}\right)\cos^2\tau d\tau$$

$$=\left(\frac{2U^2}{\pi\lambda^2}\right)\int_{\tau=0}^{\pi} \cos^2\tau d\tau$$

$$=\left(\frac{2U^2}{\pi\lambda^2}\right)\left(\frac{\pi}{2}\right)=U^2/\lambda^2 .$$

Now by x, y symmetry,

$$E(Y)=0, \text{ and}$$

$$V(Y)=U^2/\lambda^2 .$$

Let X_{Tot} be the x coordinate of the target position at time t.

Then,

$$X_{Tot}=X_1+X_2+\cdots+X_N, \qquad \text{where } N \sim \text{Poisson}(\lambda t).$$

Since variances add for independent random variables,

$$V\left(X_{Tot}\middle|N=n\right)=\sum_{i=1}^{n}V(X_i)$$

$$=n\left(U^2/\lambda^2\right).$$

Now, conditioning on N,

$$V\left(X_{Tot}\right) = \sum_{i=1}^{\infty} n\left(U^2 \big/ \lambda^2\right) P\left(N = n\right)$$

$$= \left(U^2 \big/ \lambda^2\right) E\left(N\right)$$

$$= \left(U^2 \big/ \lambda^2\right) \lambda t$$

$$= \left(U^2 \big/ \lambda\right) t \, .$$

Thus,

$$V\left(X_{Tot}\right) = \left(U^2 \big/ \lambda\right) t \, ,$$

$$V\left(Y_{Tot}\right) = \left(U^2 \big/ \lambda\right) t \, ,$$

$E\left(X_{Tot}\right) = 0$, and

$$E\left(Y_{Tot}\right) = 0 \, .$$

And by the Central Limit Theorem, after a sufficiently large number of turns, we can approximate the target's position with a circular, bivariate normal random variable with mean $\left(0,0\right)$ and variances in the x and y directions of $U^2 t \big/ \lambda$.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. ANALYTICAL APPROACHS TO FIND GOOD TRACK SPACINGS

A. S^*

Let $A(t)$ be the area enclosed by the track by time t. So,

$$A(t) \approx SVt\,.$$

Assuming that the area enclosed is approximately circular,

$$A(t) \approx \pi r^2(t)\,.$$

Equating these expressions gives

$$r(t) \approx \sqrt{SVt/\pi}, t > 0\,. \tag{1}$$

As illustrated in Figure 5, we can use these expressions to find the track spacing (S^*), such that $r(t)$ is just tangent to $U(t+\tau)-R$ at some time (t^*).

Referring to Figure 5, at time $t = t^*$,

$$U(t+\tau)-R = r(t) = \sqrt{S^* Vt/\pi}\,.$$

So,

$$\begin{aligned}
\frac{SVt}{\pi} &= \left(U(t+\tau)-R\right)^2 \\
&= \left(Ut + (U\tau - R)\right)^2 \\
&= U^2t^2 + 2U(U\tau - R)t + (U\tau - R)^2\,.
\end{aligned}$$

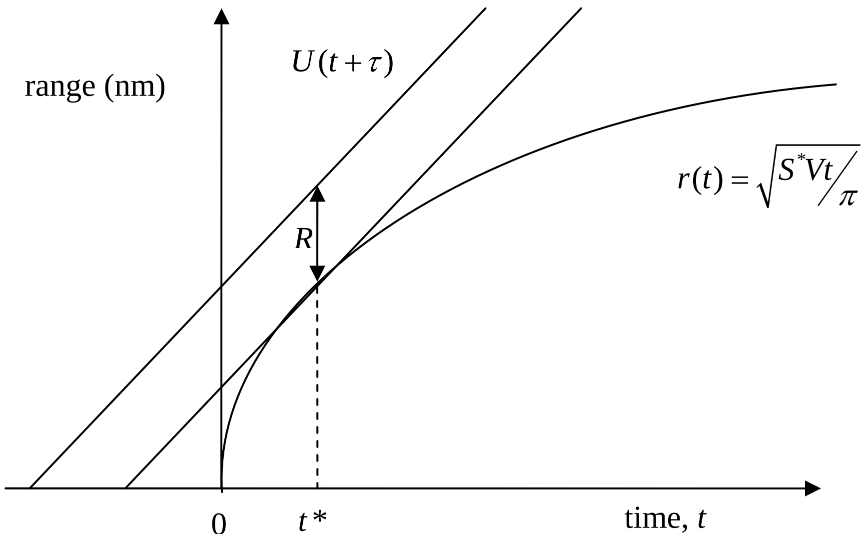


Figure 5. Calculation of S^*

Collecting terms,

$$\underbrace{U^2t^2}_{a}+\underbrace{\left(2U\left(U\tau-R\right)-SV/\pi\right)}_{b}t+\underbrace{\left(U\tau-R\right)^2}_{c}=0\,.$$

This is a quadratic equation in t. There can be 0, 1 or 2 solutions. We are looking for the S^* that produces one solution. From the quadratic formula, $4ac=b^2$ yields one solution.

So,

$$\begin{aligned}4U^2\left(U\tau-R\right)^2 &= \left(2U\left(U\tau-R\right)-\left(SV\right)/\pi\right)^2\\ &= 4U^2\left(U\tau-R\right)^2-\frac{4U\left(U\tau-R\right)SV}{\pi}+\left(SV/\pi\right)^2\,.\end{aligned}$$

One solution is $S=0$. Assuming that $S\neq 0$, and then dividing both sides by S, gives

$$\frac{-4\pi U\left(U\tau-R\right)V}{\pi}+\frac{SV^2}{\pi^2}=0\,.$$

Solving for S gives

$$S^* = \frac{4\pi U\left(U\tau - R\right)}{V}. \qquad (2)$$

S^* is the largest track spacing which keeps all search effort on the furthest-on-disk.

B. S_{Ray}

At time $t = -\tau$ (initial detection), the target's location is circular normal with means (0, 0) and variances $\left(\sigma^2, \sigma^2\right)$. The target then begins a Random Tour with parameters λ and U (target's speed).

By the Central Limit Theorem, the target position at time $t \geq -\tau$ (for sufficiently large *t*) is still circular normal with mean (0, 0), but with variances increased from σ^2 (at time $t = -\tau$) to $\sigma^2 + \frac{U^2\left(t+\tau\right)}{\lambda}$ (at time $t \geq -\tau$). And since the radial position of a circular bivariate normal random variable with variances s^2 is Rayleigh distributed with mode *s*, we know that the radial position at time $t, R(t)$, is approximately Rayleigh with mode $\sqrt{\sigma^2 + \frac{U^2\left(t+\tau\right)}{\lambda}}$.

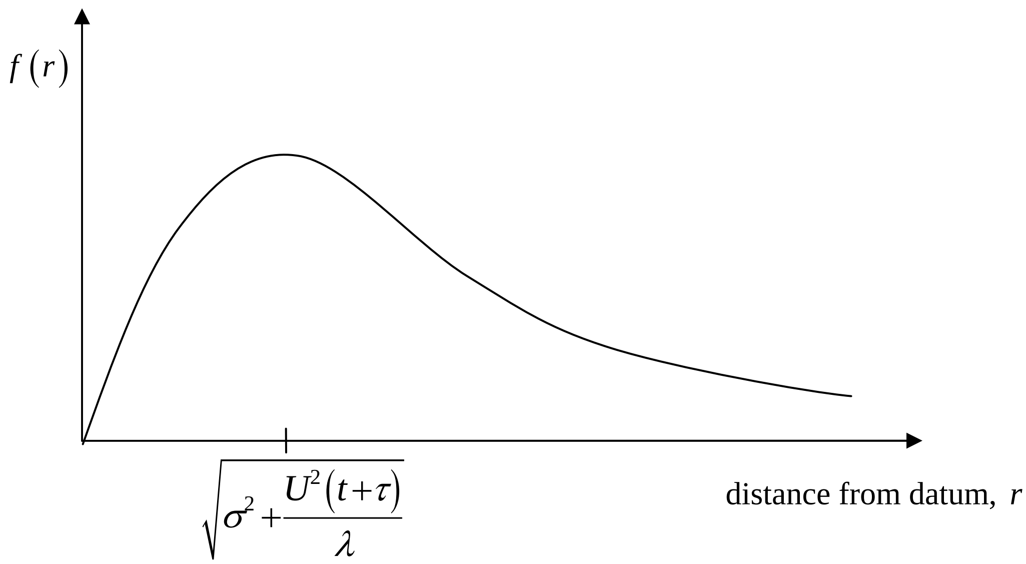


Figure 6. Density Function of Radial Position of Target at Time t, Assuming a Normal Approximation to the Random Tour

So, a spiral search which maintains a radial distance at search time t of $\sqrt{\sigma^2 + \frac{U^2(\tau+t)}{\lambda}}$, $t \geq 0$ should be a good search. The justification is that this radial distance is the mode of the approximate radial distance density function.

The Archimedes spiral search which provides the tightest support below for this function is

$$r_{Arch}(t) = \sqrt{\frac{U^2 t}{\lambda}},$$

as illustrated in Figure 7.

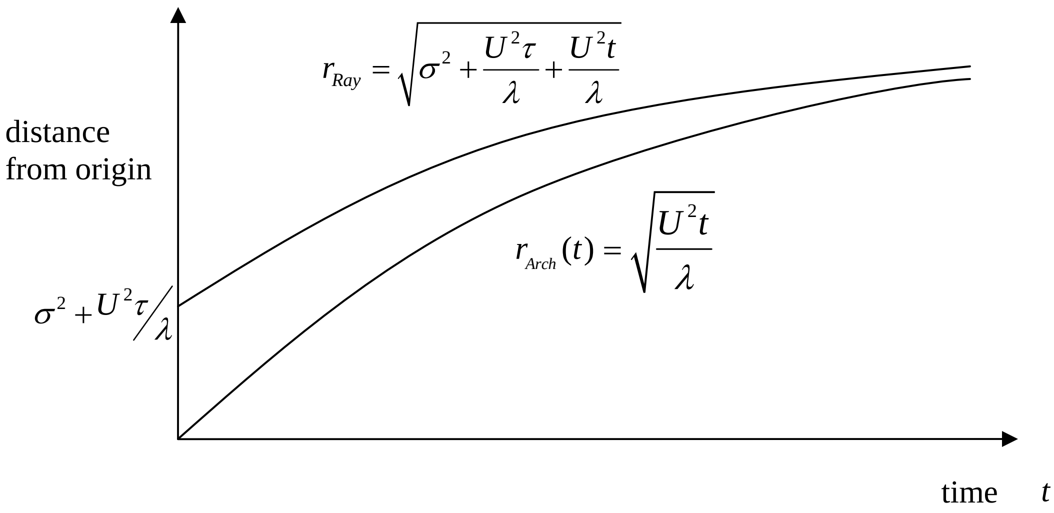


Figure 7. Calculation of S_{Ray}

Setting $\sqrt{\frac{U^2 t}{\lambda}}$ equal to $\sqrt{\frac{SVt}{\pi}}$ (from Equation (1)) and solving for S yields

$\frac{SVt}{\pi} = \frac{U^2 t}{\lambda}$, so

$$S_{Ray} = \frac{U^2 \pi}{V\lambda}. \tag{3}$$

S_{Ray} should be a good track spacing for a random tour target that is approximately normally distributed. However, a small λ can make S_{Ray} unreasonably large because the normal approximation may not apply in that case. For instance, when $U = 10\text{kts}, V = 200\text{kts}, \lambda = 0.0001/\text{hour}$, S_{Ray} is $\frac{10^2 \times \pi}{200 \times 0.0001} \approx$ 15,700 miles, which is too large.

C. $S = 2R$

When the target is slow or the searcher is fast (i.e., U/V is small), then an exhaustive search (track spacing of 2*R*) should be effective. 2*R* is the largest track spacing which guarantees no gaps in search coverage.

D. PROPOSED EXPRESSION

We have introduced three analytical expressions for the recommended track spacing : S_{Ray}, S^*, and 2*R*. Each has its weaknesses and strengths. In particular,

- S_{Ray} should be good when enough time has elapsed to allow the Central Limit Theorem to guarantee that the target distribution is roughly bivariate normal. But as we have seen, when λ is small, S_{Ray} can become arbitrarily large. Consequently, it should be bounded above by another expression, and we will use S^* for this purpose.
- S^* is intended to keep all search effort within the furthest-on-disk. As such, S^* should be (approximately) the largest track spacing considered. Otherwise search effort would be placed in regions where the target can not be located.
- 2*R* is the largest track spacing which guarantees no gaps in search coverage. Thus 2*R* should be (approximately) the smallest track spacing considered for a stationary or slow target. It might, however, be too small when *R* is small or the speed ratio U/V is large.

Considering the strengths and weaknesses of S_{Ray}, S^*, and 2*R*, the following expression is proposed for the recommended track spacing:

$$\hat{S} = \max\left\{\alpha R, \min\left\{\beta S_{Ray}, S^* + \gamma\right\}\right\}. \qquad (4)$$

Simulation will be used to determine best-fit values of α, β, and γ. The need for γ was discovered when conducting the simulation experiments. It was noted that as the initial observation error standard deviation, σ, increased, the best-fit γ also increased.

V. DESIGN OF EXPERIMENT

A. NOLH (NEARLY ORTHOGONAL LATIN HYPERCUBE) DESIGN

Simulation experiments were conducted to find a best-track spacing for the Archimedes spiral search pattern. For the design of the experiment, a full factorial design for the decision variable S was crossed with a NOLH [5] design for the other five factors. We refer to the factors other than S as "Blocking Factors". They affect the outcomes, and we wish to understand their impact, but we have no ability to change them in a real-world scenario to achieve a better outcome. To construct the NOLH, a Microsoft Excel spreadsheet created by Professor Susan Sanchez (Sanchez, 2005) was used. The number of full factorial design points for S is 181 ranging from 0 to 45nm in .25nm increments, and the number of NOLH design points is 65. Appendix A is an extract of the NOLH design which depicts the 65 design points. The 65 design points are close to orthogonal. (All pairwise correlation values for design columns are lower than 0.05.) The two designs are crossed to create the following design table. The number of crossed design points is $65 \times 181 = 11{,}765$.

DP	U	τ	V	R	λ
1	14.656	0.57	153.91	0.992	0.25
2	19.109	1.578	116.41	1.133	0.688
3	17.922	1.039	242.97	0.828	0.594
...	...	...	...	...	...
65	3.375	1.016	109.38	0.711	0.563

NOLH Design for Five Blocking Factors

$\times$

DP	S
1	0
2	0.25
...	...
181	45

Full Factorial Design for Decision Variable *S*

$=$

DP	U	τ	V	R	λ	S**=0**	S**=0.25**	**...**	S**=45**
1	14.656	0.57	153.91	0.992	0.25				
2	19.109	1.578	116.41	1.133	0.688				
3	17.922	1.039	242.97	0.828	0.594				
…	...	...	...	...	...				
65	3.375	1.016	109.38	0.711	0.563				

Crossed Design

Figure 8. Crossed Design for Six Factors

B. THE MAXIMUM PROBABILITY OF DETECTION (Pd_{best}) FOR EACH DESIGN POINT

Appendix B is the JAVA code for the search simulation model used for data collection. One thousand replications are used for each of the 11,765 crossed design points. By running the simulation model, the probability of detection (Pd) for each design point was estimated. Table 1 and Figure 9 show some of the Pd results.

Blocking Factor DP	***U***	τ	***V***	***R***	λ	***Pd***			
						***S*=0**	***S*=0.25**	**...**	***S*=45**
1	14.656	0.57	153.91	0.992	0.25	**0**	**0.054**	**...**	**0.08**
2	19.109	1.578	116.41	1.133	0.688	**0**	**0.061**	**...**	**0.078**
3	17.922	1.039	242.97	0.828	0.594	**0**	**0.089**	**...**	**0.038**
...	...	...	...	...	...	**...**	**...**	**...**	**...**
64	7.531	1.813	144.53	1.156	0.094	**0**	**0.041**	**...**	**0.069**
65	3.375	1.016	109.38	0.711	0.563	**0**	**0.397**	**...**	**0.11**

Table 1. Probability of Detection (Pd) for Each Blocking Factor Design Point

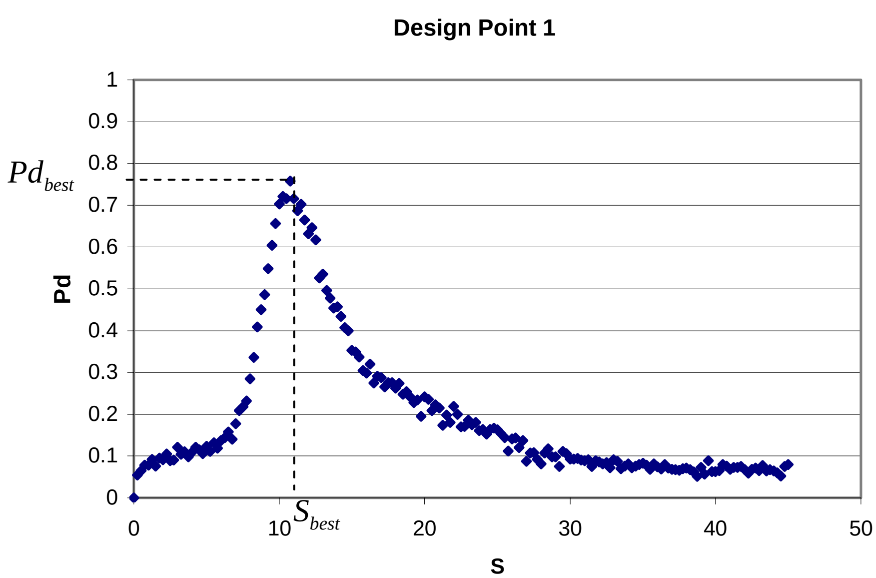


Figure 9. Plot of Estimated Simulation Probabilities of Detection for Blocking Factor Design Point 1

Figure 9 shows the simulation *Pd* for design point 1 and the 181 track spacings (0-45nm). We can now determine the largest *Pd* (Pd_{best}) and the track spacing producing Pd_{best} (S_{best}). Example results are given in Figure 9 and Table 2.

Blocking Factor DP	U	τ	V	R	λ	S_{best}	Pd_{best}
1	14.656	0.57	153.91	0.992	0.25	**10.75**	**0.758**
2	19.109	1.578	116.41	1.133	0.688	**22.5**	**0.116**
3	17.922	1.039	242.97	0.828	0.594	**17.75**	**0.39**
...	...	...	...	...	...	**...**	**...**
64	7.531	1.813	144.53	1.156	0.094	**9.5**	**0.835**
65	3.375	1.016	109.38	0.711	0.563	**1.75**	**0.943**

Table 2. Table for S_{best} and Pd_{best} for Each Blocking Factor Design Point

C. FINDING THE BEST α, β, AND γ

Using the simulation data of Tables 1 and 2, we can determine the values of α, β, and γ in Equation (4) which produce a track spacing and resulting *Pd* as close as possible to the maximum obtainable *Pd*. We use the following procedure:

For each of the 51^3 possible values of $(\alpha,\ \beta,\ \gamma)$, where α, β and γ vary over $\{0,\ .1,\ .2,\ ...,\ 5\}$

For each of the 65 blocking factor design points (dp)

Determine $\hat{S}(\alpha,\ \beta,\ \gamma,\ \mathrm{dp})$ from Equation (4)

Determine $\hat{P}d\left(\hat{S},\ \mathrm{dp}\right)$ from Table 1

Determine $Pd_{best}(\text{dp})$ from Table 2

Determine $\text{MOE}_1(\alpha, \beta, \gamma, \text{dp}) = \hat{P}d \big/ Pd_{best}$

Continue for each of the design points. Then compute

$$\text{MOE}_2(\alpha,\ \beta,\ \gamma) = \left(\frac{1}{65}\right)\sum_{\forall \text{dp}} \text{MOE}_1(\alpha,\ \beta,\ \gamma,\ dp)$$

Continue for next $(\alpha,\ \beta,\ \gamma)$ to find by enumeration

$$(\alpha^*,\ \beta^*,\ \gamma^*) = \underset{\alpha,\beta,\gamma}{\arg\max}\ \ \text{MOE}_2(\alpha, \beta, \gamma)$$

For each $(\alpha,\ \beta,\ \gamma)$ triplet, the analysis generates 65 values of $\text{MOE}_1(\alpha,\ \beta,\ \gamma,\ \text{dp})$ – one for each of the blocking factor design points. Then $\text{MOE}_2(\alpha,\ \beta,\ \gamma)$ is the average of these 65 values. $(\alpha^*, \beta^*, \gamma^*)$ are the values which maximize the average of the 65 values of MOE_1.

VI. SIMULATION RESULTS

A. MAIN RESULT

Using the enumeration procedure of the previous section, $51^3\ (=132,651)$ values of $\mathrm{MOE}_2(\alpha,\ \beta,\ \gamma)$ were computed. The largest MOE_2 was .928, and the maximizing arguments were $(\alpha^*, \beta^*, \gamma^*) = (2,\ 2.7,\ 1.8)$.

B. VISUALIZATION OF MAIN RESULT USING CONTOUR PLOTS

Contour plots were constructed to help visualize how $\mathrm{MOE}_2(\alpha,\ \beta,\ \gamma)$ varied over the range of its arguments. These plots reproduces below indicated that the best values of $(\alpha,\ \beta,\ \gamma)$ were approximately $(2,\ 2.7,\ 1.8)$.

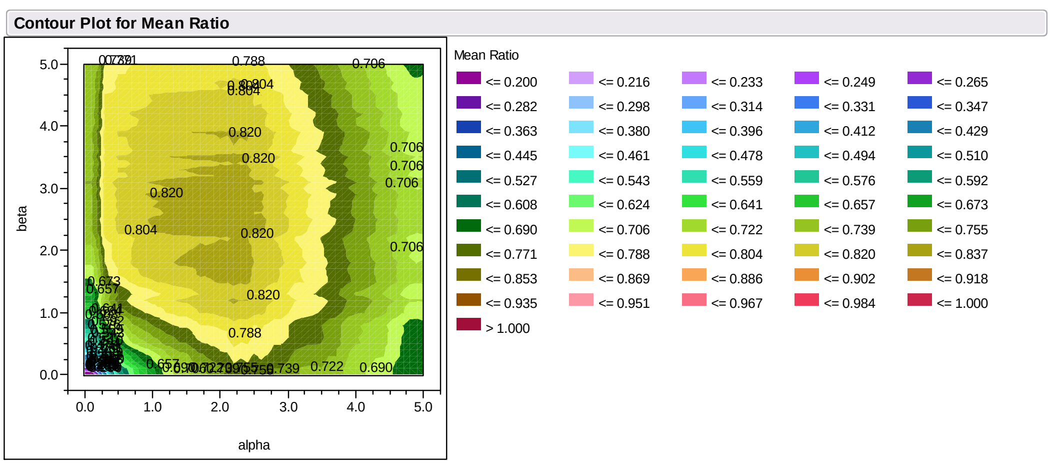


Figure 10. Contour Plot for α and β given $\gamma = 0$

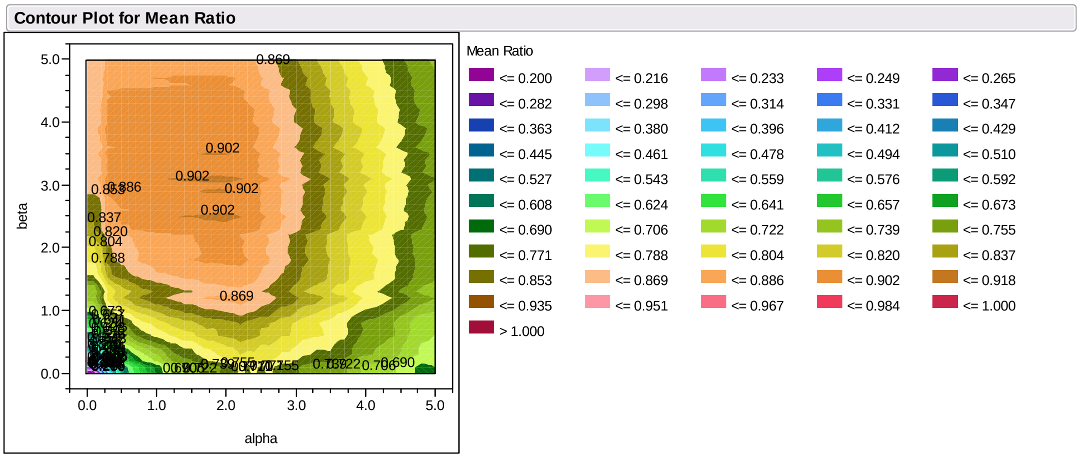


Figure 11. Contour Plot for α and β given $\gamma = 1$

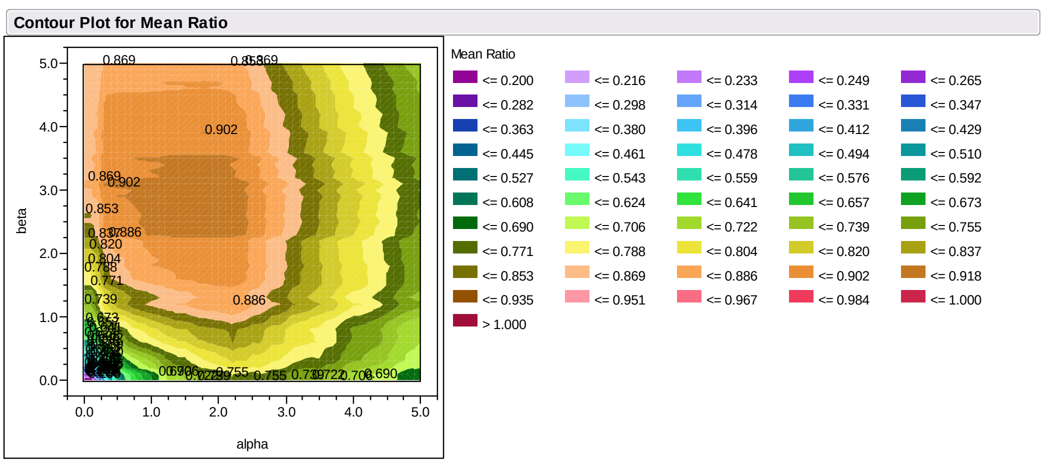


Figure 12. Contour Plot for α and β given $\gamma = 2$

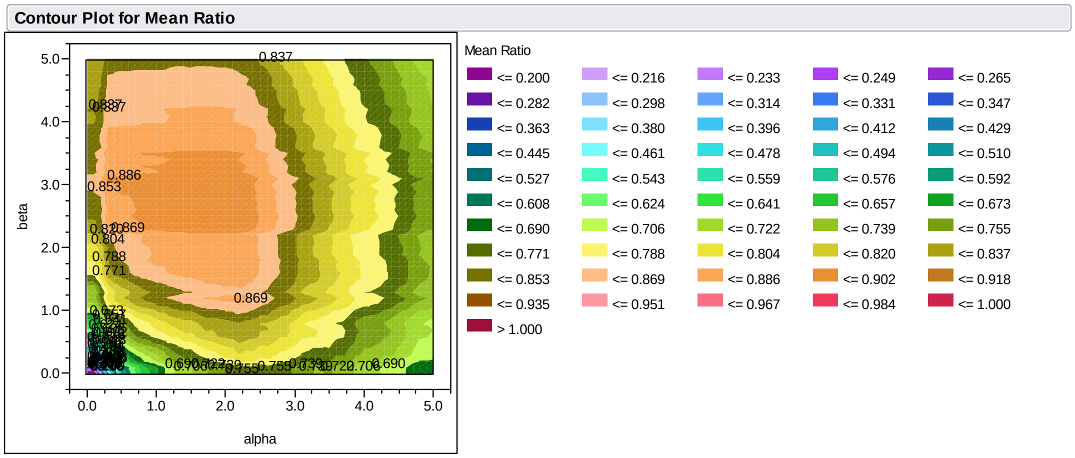


Figure 13. Contour Plot for α and β given $\gamma = 3$

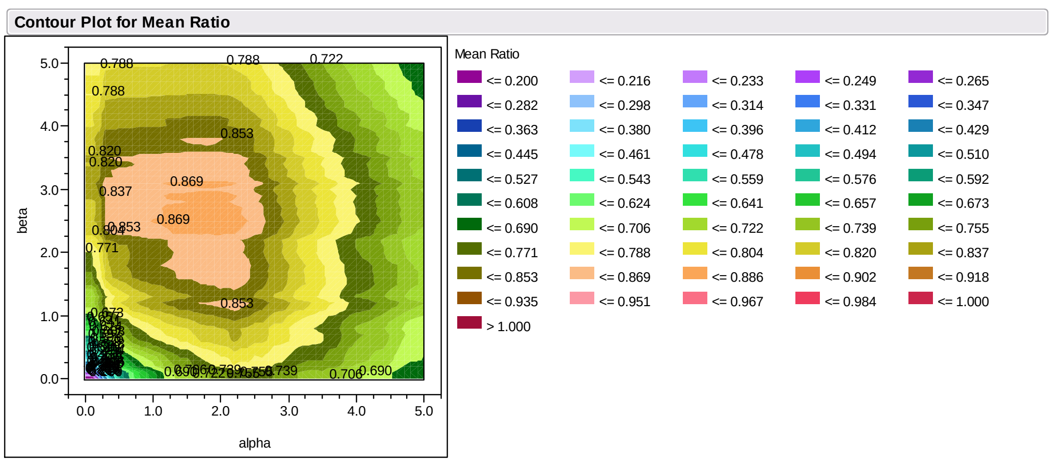


Figure 14. Contour Plot for α and β given $\gamma = 4$

THIS PAGE INTENTIONALLY LEFT BLANK

VII. CONCLUSION / FUTURE RESEARCH

A. CONCLUSION

Since the military threat of hostile submarines is increasing, the need for effective ASW operations is also increasing. In this thesis, the recommended track spacing for an Archimedes spiral search in a datum search problem was studied. This thesis has sought a meta-model to recommend a good track spacing. This analysis combined three analytical functions into a single parameterized expression. To find the best-fit parameters maximizing the average ratio $\hat{P}d \big/ Pd_{best}$, a simulation experiment with a NOLH (Nearly Orthogonal Latin Hypercube) design was used. The complete model is

$$\hat{S} = \max\{2R, \min\{2.7S_{ray}, S^* + 1.8\}\},$$

where R is detection range, S^* and S_{Ray} are given by Equations (2) and (3).

B. FUTURE RESEARCH

This thesis blended different techniques involving stochastic modeling and simulation, design of experiments, and optimization to create a meta-model for track spacing for an Archimedes Spiral. Suggested future research includes:

- Good search patterns with multiple targets and multiple searchers;
- Selection of appropriate spiral search pattern in datum search problem;
- Factor analysis for probability of detection in ASW datum search;
- Recommended track spacing when the other parameters are unknown;
- Impact on observation error for probability of detection.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A. NOLH (NEARLY ORTHOGONAL LATIN HYPERCUBE) DESIGN FOR FIVE BLOCKING FACTORS

Low level	**1**	**0.5**	**100**	**0.5**	**0**
High level	**20**	**2**	**250**	**2**	**2**
decimals	**3**	**3**	**2**	**3**	**3**
factor name	$\boldsymbol{U}$	τ	$\boldsymbol{V}$	$\boldsymbol{R}$	λ
1	14.656	0.57	153.91	0.992	0.25
2	19.109	1.578	116.41	1.133	0.688
3	17.922	1.039	242.97	0.828	0.594
4	13.172	1.836	207.81	1.18	0.125
5	18.516	1.203	128.13	0.523	0.188
6	11.094	1.883	135.16	1.227	0.313
7	15.547	0.781	179.69	0.547	0.5
8	16.438	1.648	238.28	0.969	0.75
9	14.063	0.547	102.34	1.719	0.813
10	19.406	1.531	172.66	1.648	0
11	10.797	0.523	245.31	1.297	0.406
12	19.703	1.273	203.13	1.883	0.344
13	11.391	0.828	149.22	1.461	0.781
14	15.844	1.32	163.28	1.836	0.531
15	11.984	0.992	217.19	1.93	0.875
16	13.766	1.367	182.03	1.578	0.281
17	17.328	1.156	158.59	0.688	1.344
18	11.688	1.789	139.84	1.016	1.156
19	14.953	1.086	184.38	0.945	1.563
20	12.578	1.602	228.91	0.594	1.094
21	16.734	0.805	160.94	0.641	1.969
22	16.141	1.742	100	0.734	1.031
23	20	1.109	219.53	0.875	1.938
24	12.281	2	224.22	1.063	1.375
25	12.875	0.875	114.06	1.391	1.625
26	15.25	1.438	123.44	2	1.531
27	17.031	0.641	212.5	1.695	1.063
28	18.219	1.766	193.75	1.742	1.844
29	18.813	0.945	118.75	1.414	1.281
30	14.359	1.906	151.56	1.602	1.781
31	13.469	0.688	205.47	1.344	1.906
32	17.625	1.484	240.63	1.789	1.438
33	10.5	1.25	175	1.25	1
34	6.344	1.93	196.09	1.508	1.75
35	1.891	0.922	233.59	1.367	1.313
36	3.078	1.461	107.03	1.672	1.406
37	7.828	0.664	142.19	1.32	1.875
38	2.484	1.297	221.88	1.977	1.813
39	9.906	0.617	214.84	1.273	1.688
40	5.453	1.719	170.31	1.953	1.5

Low level	1	0.5	100	0.5	0
High level	**20**	**2**	**250**	**2**	**2**
decimals	**3**	**3**	**2**	**3**	**3**
factor name	***U***	τ	***V***	***R***	λ
41	4.563	0.852	111.72	1.531	1.25
42	6.938	1.953	247.66	0.781	1.188
43	1.594	0.969	177.34	0.852	2
44	10.203	1.977	104.69	1.203	1.594
45	1.297	1.227	146.88	0.617	1.656
46	9.609	1.672	200.78	1.039	1.219
47	5.156	1.18	186.72	0.664	1.469
48	9.016	1.508	132.81	0.57	1.125
49	7.234	1.133	167.97	0.922	1.719
50	3.672	1.344	191.41	1.813	0.656
51	9.313	0.711	210.16	1.484	0.844
52	6.047	1.414	165.63	1.555	0.438
53	8.422	0.898	121.09	1.906	0.906
54	4.266	1.695	189.06	1.859	0.031
55	4.859	0.758	250	1.766	0.969
56	1	1.391	130.47	1.625	0.063
57	8.719	0.5	125.78	1.438	0.625
58	8.125	1.625	235.94	1.109	0.375
59	5.75	1.063	226.56	0.5	0.469
60	3.969	1.859	137.5	0.805	0.938
61	2.781	0.734	156.25	0.758	0.156
62	2.188	1.555	231.25	1.086	0.719
63	6.641	0.594	198.44	0.898	0.219
64	7.531	1.813	144.53	1.156	0.094
65	3.375	1.016	109.38	0.711	0.563

APPENDIX B. JAVA CODE FOR DATA COLLECTION

The Java code within this appendix collects all the simulation results over 11765 crossed design points.

```
/*
 * File: Arch.java
 * Created on Sep 15, 2007, 12:00 PM
 */

/**
        * @author Byungsoo Son
        * Class to collect the probability of detection
        * for the 11765 design points
        */
import java.util.Random;


public class Arch {


       public static void main(String[] args) {
              //set input values
              Random generator = new Random();
              //initializing all variables
              double targetSpeed=0;
              double tau=0;
              double tMax =5;
              double searcherSpeed=0;
              double detectionRange=0;
              double trackSpacing=0;
              double lambda=0;
              double timeStep=0;
              double xTarget=0;
              double yTarget=0;
              double sigma=0;
              double xSearcher=0;
              double ySearcher=0;
              double distance=0;
              int Detection=0;
              double courseDirection;
              int numberRelications = Integer.parseInt(args[0]);
              int numberDetection=0;

              if (args.length == 9){
                     sigma = Double.parseDouble(args[1]);
                     targetSpeed = Double.parseDouble(args[2]);
                     tau = Double.parseDouble(args[3]);
                     searcherSpeed = Double.parseDouble(args[4]);
                     detectionRange = Double.parseDouble(args[5]);
                     lambda = Double.parseDouble(args[6]);
                     trackSpacing = Double.parseDouble(args[7]);
                     timeStep = Double.parseDouble(args[8]);
              }
              else {
                        System.err.println("Supply nine arguments: " +
                                    "numberReplications(int), sigma(double) " +
                                    "targetSpeed(double), tau(double), " +
```

```
                    "searcherSpeed(double), " +
                    "detectionRange(double), lambda(double), " +
                    "trackSpacing(double) " +
                    "and timeStep(double)");
            System.exit(-1);
        }
        // replicate n times given the other parameters
        for(int n=0;n<numberReplications;n++){
                xTarget = generator.nextGaussian()*sigma;
                yTarget= generator.nextGaussian()*sigma;
                xSearcher=0;
                ySearcher=0;
                Detection=0;
                courseDirection =
                        generator.nextDouble()*2*Math.PI;
                //target motion model for time -tau to 0
            for(int i=0;i<(tau/timeStep);i++){
                if(generator.nextDouble()<lambda*timeStep){
                    courseDirection =
                        generator.nextDouble()*2*Math.PI;
                }
                xTarget = xTarget + Math.cos(courseDirection)
                                *targetSpeed*timeStep;
                yTarget = yTarget + Math.sin(courseDirection)
                                *targetSpeed*timeStep;
            }
            //searcher and target motion model for time 0 to tMax
            //If the distance between searcher and target is less
            //than the detection range of sensor,detection occurs
            for(int j=0;j<(tMax/timeStep);j++){
                if(generator.nextDouble()< lambda*timeStep){
                    courseDirection = generator.nextDouble()
                    *2*Math.PI;
                    }
                xTarget = xTarget + Math.cos(courseDirection)
                                * targetSpeed*timeStep;
                yTarget = yTarget + Math.sin(courseDirection)
                                * targetSpeed*timeStep;
                xSearcher = Math.sqrt(trackSpacing * searcherSpeed
                                * (timeStep * j)/Math.PI)
                                * Math.cos(2*Math.sqrt(searcherSpeed
                                *Math.PI* timeStep*j/trackSpacing));
                ySearcher = Math.sqrt(trackSpacing * searcherSpeed
                                * (timeStep * j)/Math.PI)
                                * Math.sin(2*Math.sqrt(searcherSpeed
                                *Math.PI * timeStep*j/trackSpacing));
                distance = Math.sqrt(Math.pow(xTarget-xSearcher,2)
                                + Math.pow(yTarget-ySearcher,2));
                if(distance < detectionRange){
                    Detection = Detection + 1;
                    break;
                }
            }
            numberDetection = numberDetection + Detection;
        }
        System.out.println((double)numberDetection/numberReplications);
    }
}
```

LIST OF REFERENCES

[1] Daniel Zwillinger, CRC Standard Mathematical Tables and Formulae, 31st ed., Chapman & Hall/CRC, 2003, 343 pp.

[2] Alan Washburn, Probability Density of a Moving Particle, The Boeing Company, Seattle, WA, 1967.

[3] Sheldon M. Ross, Introduction to Probability Models 8th ed., San Diego, CA: Academic Press, 2003, pp. 241-289.

[4] Jay L. Devore, Probability and Statistics for Engineering and the Science, 6th ed., Thomson Brooks/Cole, 2004, pp. 239-242.

[5] Susan M. Sanchez, Work Smarter, Not Harder: Guidelines for Designing Simulation Experiments, Proceedings of the 2006 Winter Simulation Conference, 2006.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. James Eagle
   Naval Postgraduate School
   Monterey, California

4. Paul Sanchez
   Naval Postgraduate School
   Monterey, California

5. Jae-Young Lee
   Korea National Defense University
   Seoul, Republic of Korea